

PERENCANAAN DAN PEMBUATAN APLIKASI GAME TEBAK NADA PADA OS ANDROID

Dekris Darutama, Reni Soelistijorini, B.Eng, M.T, Achmad Subhan KH.ST

Mahasiswa Jurusan Teknik Telekomunikasi
Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember, Kampus ITS, Surabaya, 60111
email : dekris.434@gmail.com

ABSTRAK

Kepekaan terhadap suara nada – nada dari tangga nada – tangga nada dasar dirasa kurang dalam realita kehidupan saat ini. Lebih cenderung mendengarkan daripada mempelajari nada – nada apa yang membuat suatu lagu dapat diputar merupakan tradisi yang dibudayakan. Karenanya mendengarkan dan memahami nada – nada perlu dijadikan satu kemasan yang baik dan tepat sasaran. Dalam mendukung kemasan yang tepat sasaran, perlu dikembangkan suatu aplikasi yang mampu membungkus kemasan tersebut menjadi menarik. Android merupakan salah satu target dalam menembak pemasaran aplikasi ini. Pemrograman yang digunakan yaitu pemrograman dengan berkilat pemrograman java dengan didukung oleh SDK Android sebagai emulator dan Eclipse sebagai *java development tool*. Nada – nada yang dimainkan berupa file midi yang mana file midi lebih jernih didengar dan lebih kecil dari sisi ukuran file dibandingkan lainnya. Aplikasi ini pada awal mulanya pemain diminta dan mendengarkan dan menekan karakter yang ditampilkan. Disisi lain terdapat pengenalan terhadap nada – nada dasar. Dari sistem yang dibuat menghasilkan aplikasi yang mampu memainkan nada – nada midi dan mendapatkan suatu nilai *frame per seconds* (fps).

Kata Kunci : Java Sound, Tone Generator dan Synthesizer Library

1. PENDAHULUAN

Seiring dengan berkembang dan pesatnya teknologi aplikasi pada perangkat-perangkat telekomunikasi. Ditambah lagi dengan perangkat-perangkat telekomunikasi yang bergerak cepat dalam memperbarui sistem operasi. Android merupakan sistem operasi open source yang dimana merupakan pilihan dalam memenuhi kebutuhan tersier dalam kehidupan.

Musik merupakan hal yang tidak terlepas dari kehidupan masyarakat. Tua muda pun gemar berinteraksi dalam musik. Masyarakat lebih tertarik mendengarkan musik saat mereka menemukan musik yang mereka favoritkan. Namun masyarakat kurang tertarik dalam memahami bahwa dari nada – nada dasar apa lagu itu berasal. Untuk itu, agar animo masyarakat bertambah dalam mengetahui nada – nada dasar, dibutuhkan beberapa cara untuk menarik masyarakat. Salah satunya adalah Aplikasi Tebak Nada pada OS Android.

Untuk mempermudah masyarakat dalam mengenali nada – nada dasar, maka aplikasi ini menggunakan format nada midi. Ini didasari oleh antusias masyarakat yang menggunakan aplikasi *composer* pada ponsel – ponsel yang masih menggunakan *polyphonic*. Aplikasi *composer* tersebut menggabungkan nada – nada dasar sehingga menjadi lagu dalam format midi.

Ponsel bersistem operasi Android yang mempunyai layar yang berukuran besar dan

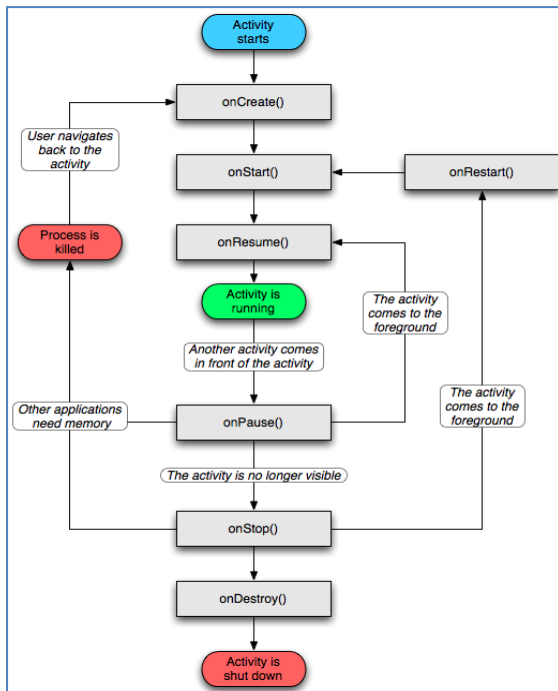
touchscreen merupakan pendukung dari aplikasi ini, sehingga aplikasi ini berjalan secara maksimal.

2. LANDASAN TEORI

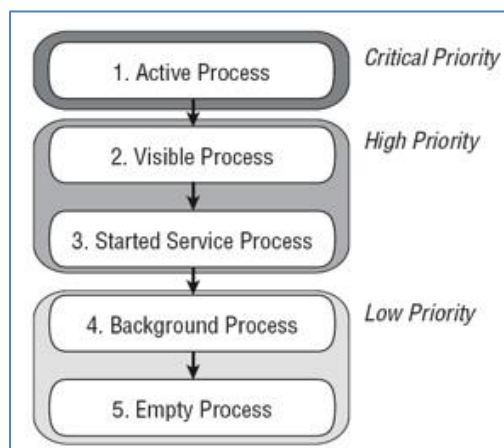
2.1 ANDROID

Android adalah sistem operasi untuk smartphone yang dibuat oleh google corporation. Sistem operasi ini dikembangkan dengan memanfaatkan linux kernel. Android mempunyai siklus hidup yang merupakan logika dasar aliran dari sebuah aplikasi yang dibangun. Sistem sangat berperan menentukan apakah aplikasi dijalankan, dihentikan sementara atau dihentikan sama sekali. Jika pengguna menggunakan sebuah *Activity*, maka sistem akan memberikan prioritas utama untuk aplikasi tersebut. Sebaliknya jika suatu *Activity* tidak terlihat dan sistem membutuhkan sumber daya yang lebih, maka *Activity* prioritas rendah akan ditutup. Android menjalankan setiap aplikasi dalam proses secara terpisah, yang masing – masing memiliki mesin virtual pengolah sendiri, sengan ini melindungi penggunaan memori pada aplikasi. Selain itu Android dapat mengontrol aplikasi mana yang layak menjadi prioritas utama. Karena Android sangat sensitif dengan siklus hidup aplikasi dan komponen – komponennya. Perlu adanya penanganan terhadap setiap kondisi agar aplikasi menjadi stabil. Untuk komponen aplikasi activity diagram aliran program tampak pada

gambar 2.1 Sedangkan gambar 2.2 menunjukkan prioritas dari aplikasi



Gambar 2.1 Diagram alur siklus Android

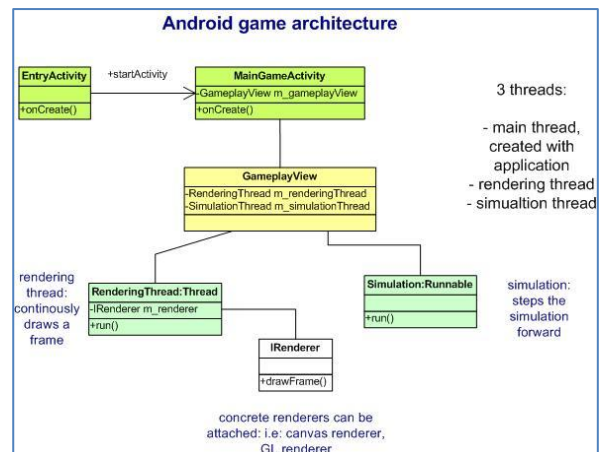


Gambar 2.2 Prioritas aplikasi berdasarkan *activity*

2.2 ANDROID GAME ARSITEKTUR

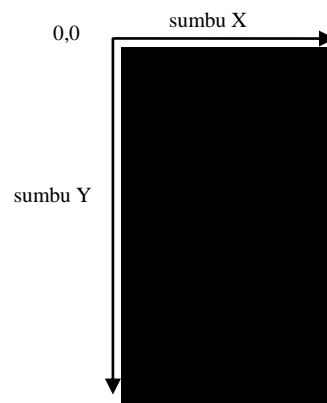
Arsitektur dalam game Android memiliki tiga komponen utama, yaitu *main thread*, *rendering thread* dan *simulation thread*.

Main thread dibuat setiap aplikasi dibuat dimana *main thread*. *Rendering thread* selalu menggambarkan frame dan merupakan *thread* yang independen dari *simulation*. *Simulation thread* mengawasi kinerja dari waktu, *update game word*, kalkulasi matematis dan lain sebagainya. Jika terlalu lama dalam memanggil animasi yang akan ditampilkan, *simulation* tidak akan menunggu sampai animasi *finish* tetapi akan melanjutkan kinerjanya sendiri. Berikut adalah gambar arsitektur dari game pada Android.



Gambar 2.3 Android game arsitektur.

Pada *game* Android memiliki suatu koordinat sumbu X dan sumbu Y. Koordinat tersebut digunakan saat suatu *game* Android melakukan *looping* sehingga mempermudah mendeklarasikan suatu fungsi.



Gambar 2.4 Koordinat layar Android

2.3 JAVA SOUND

Java Sound menyediakan sekumpulan API untuk memanipulasi *sound* atau media seperti video. API ini harus di download terlebih dahulu karena tidak langsung di bundle dalam *Java Audio SDK/SE*. Dalam *audio* sound/media digunakan untuk menghasilkan dari frekuensi menjadi suara dalam format midi.

Kebanyakan program yang menyediakan sendiri paket MIDI *Java Sound* API melakukannya untuk menyatukan suara. Keseluruhan file MIDI, peristiwa, urutan, dan *sequencers*, hampir selalu memiliki tujuan dan akhirnya pengiriman data musik untuk menyatukan dan untuk mengkonversi menjadi *audio*. (termasuk program yang mengkonversi MIDI menjadi notasi musik yang dapat dibaca oleh note balok, dan program yang mengirim pesan ke perangkat MIDI-dikendalikan eksternal seperti pencampuran konsol.)

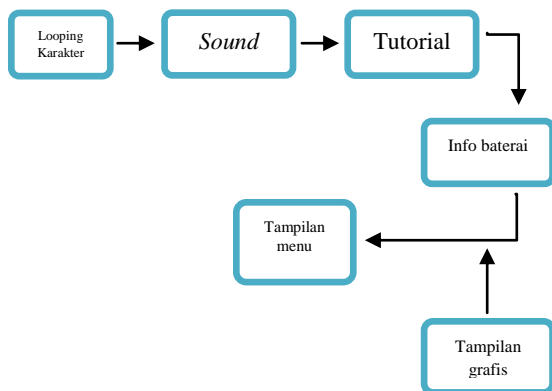
Interface dari synthesizer merupakan dasar dari pemaketan MIDI. Dan ini menunjukkan

bagaimana untuk memanipulasi *synthesizer* untuk memainkan suara. Banyak program hanya menggunakan *sequencer* untuk mengirim data MIDI file ke *synthesizer*, dan tidak perlu memanggil banyak metode *synthesizer* secara langsung.

3. METODOLOGI

3.1 Perencanaan Sistem

Pembuatan sistem dilakukan dilakukan pada sistem operasi Windows 7. Dimana membutuhkan ketersediaannya perangkat – perangkat lunak yang telah dijelaskan pada point 3.1.1 yaitu persiapan. Berikut adalah sistematis membuat sistem utama.



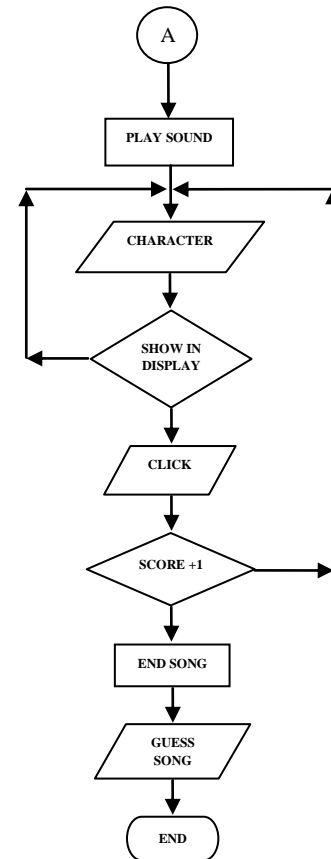
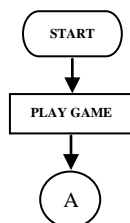
Gambar 3.1 Sistem yang dibangun

Untuk mempermudah dalam merancang dan membuat aplikasi ini dibuatlah suatu blok diagram sistem. Semua secara keseluruhan mempunyai peranan penting dalam menghasilkan suatu aplikasi game tebak nada ini.

Penjelasan sistem secara umum, seperti pada gambar diatas adalah sebagai berikut.

1. Looping Karakter dibuat untuk membuat gambar / karakter melakukan perulangan.
2. Sound dimaksud yaitu menambahkan *sound audio* pada aplikasi.
3. Tutorial dibuat untuk mengenali nada – nada.
4. Tampilan grafis membuat tampilan dengan photoshop.
5. Tampilan menu merupakan pembuatan tampilan menu secara keseluruhan.

Selanjutnya dibuat suatu flowchart sistem kerja dari aplikasi ini.



Gambar 3.2 Flowchart sistem

Dari flowchart diatas, dijelaskan mengenai sistem kinerja dari aplikasi ini. Aplikasi dimulai dengan memainkan game kemudian dimainkanlah audio sound yang diletakkan di direktori *res/raw/*. Nantinya muncul sebuah karakter secara *looping* dalam kondisi acak (*random*). Jika karakter ditekan menghasilkan nilai score yang selalu bertambah. Jika tidak ditekan pada karakter maka karakter melakukan looping terus menerus sampai dengan ditekannya menu *back* pada ponsel.

Setelah mengerjakan sistem selesai maka dikonversikan aplikasi tersebut menjadi file apk. File ini nantinya dibaca oleh ponsel berbasis sistem operasi Android untuk dijadikan aplikasi.

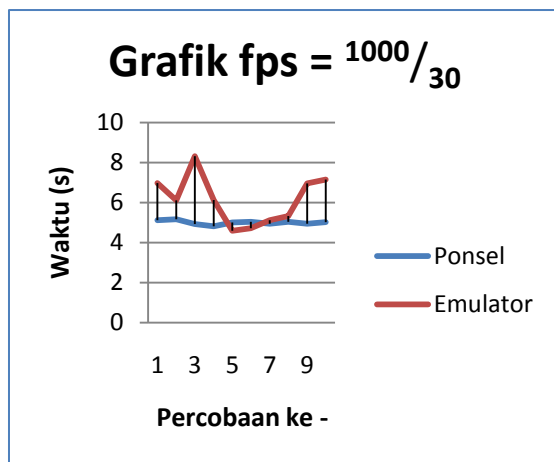
4. PENGUJIAN DAN ANALISA

4.1 Pengujian *frame per seconds*

Untuk mendapatkan waktu eksekusi karakter ini maka perlu dilakukan pengujian pengukuran waktu eksekusi. Proses pengukuran waktu eksekusi ini dimulai saat karakter muncul sampai dengan karakter menghilang. Pengukuran didasarkan besaran dari *frame per seconds*. Berikut ini adalah hasil pengujian yang telah dilakukan setelah mengambil sebanyak sepuluh kali percobaan setiap perubahan fps pada ponsel nexus one dan emulator.

Tabel 4.1 Tabel data pengukuran ($\text{fps} = 1000/30$)

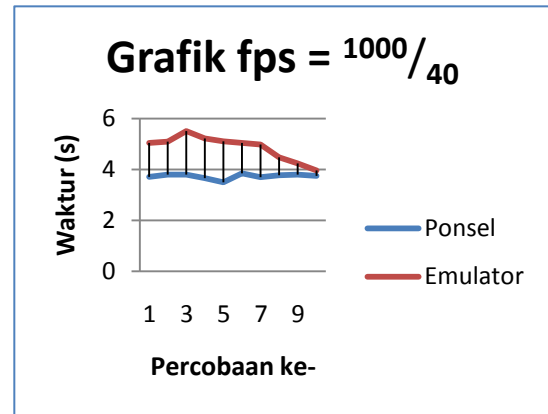
Percobaan ke -	Waktu (s)	
	Ponsel	Emulator
1	5.11	6.97
2	5.16	6.10
3	4.93	8.31
4	4.81	6.13
5	5.00	4.59
6	5.04	4.71
7	4.94	5.12
8	5.04	5.32
9	4.94	6.95
10	5.02	7.14

**Gambar 4.1** Grafik Pengujian $\text{fps} = 1000/30$

Berdasarkan pada pengukuran dengan $\text{fps} = 1000/30$ didapat waktu saat karakter muncul samapi karakter menghilang, dapat disimpulkan bahwasanya rata – rata pemain dalam mengakses aplikasi game pada ponsel selama tersebut selama 4.99s. Sedangkan pada emulator hasil rata – ratanya 6.134s.

Tabel 4.2 Tabel data pengukuran ($\text{fps} = 1000/40$)

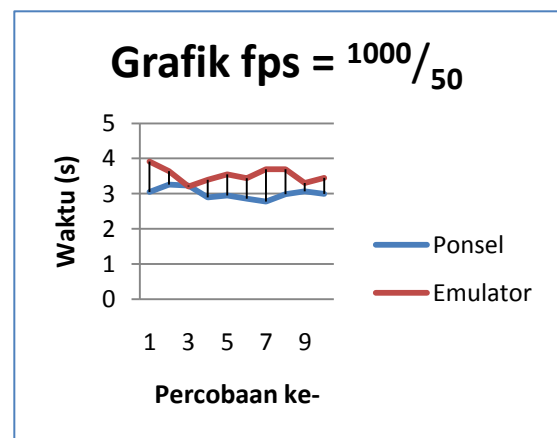
Percobaan ke -	Waktu (s)	
	Ponsel	Emulator
1	3.71	5.04
2	3.79	5.09
3	3.80	5.51
4	3.66	5.21
5	3.50	5.10
6	3.84	5.04
7	3.70	4.98
8	3.77	4.47
9	3.80	4.23
10	3.74	3.96

**Gambar 4.2** Grafik Pengujian $\text{fps} = 1000/40$

Berdasarkan pada pengukuran dengan $\text{fps} = 1000/40$ didapat waktu saat karakter muncul samapi karakter menghilang, dapat disimpulkan bahwasanya rata – rata pemain dalam mengakses aplikasi game pada ponsel selama tersebut selama 3.731. Sedangkan pada emulator hasil rata – ratanya 4.863s.

Tabel 4.3 Tabel data pengukuran ($\text{fps} = 1000/50$)

Percobaan ke -	Waktu (s)	
	Ponsel	Emulator
1	3.05	3.91
2	3.26	3.64
3	3.23	3.20
4	2.89	3.39
5	2.95	3.55
6	2.86	3.44
7	2.77	3.69
8	2.98	3.69
9	3.07	3.30
10	2.99	3.45

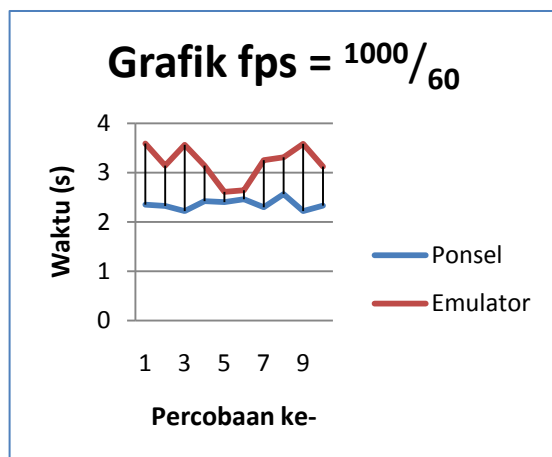
**Gambar 4.3** Grafik Pengujian $\text{fps} = 1000/50$

Berdasarkan pada pengukuran dengan $\text{fps} = 1000/50$ didapat waktu saat karakter muncul samapi

karakter menghilang, dapat disimpulkan bahwasanya rata – rata pemain dalam mengakses aplikasi game pada ponsel selama tersebut selama 3.005s. Sedangkan pada emulator hasil rata – ratanya 3.526s.

Tabel 4.4 Tabel data pengukuran ($\text{fps} = 1000/60$)

Percobaan ke -	Waktu (s)	
	Ponsel	Emulator
1	2.35	3.59
2	2.32	3.14
3	2.22	3.56
4	2.42	3.14
5	2.40	2.61
6	2.46	2.64
7	2.30	3.25
8	2.56	3.31
9	2.22	3.58
10	2.33	3.12



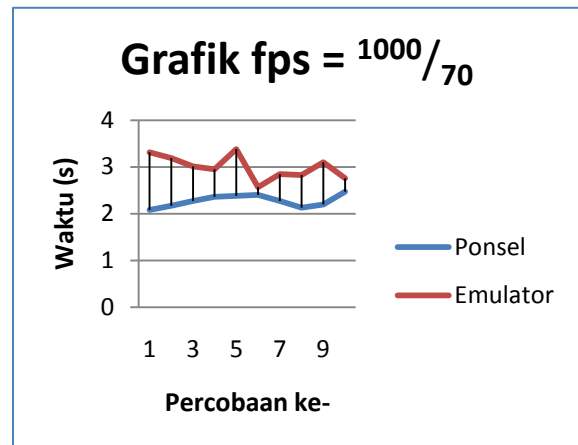
Gambar 4.4 Grafik Pengujian $\text{fps} = 1000/60$

Berdasarkan pada pengukuran dengan $\text{fps} = 1000/60$ didapat waktu saat karakter muncul samapi karakter menghilang, dapat disimpulkan bahwasanya rata – rata pemain dalam mengakses aplikasi game pada ponsel selama tersebut selama 2.358s. Sedangkan pada emulator hasil rata – ratanya 3.194s.

Tabel 4.5 Tabel data pengukuran ($\text{fps} = 1000/70$)

Percobaan ke -	Waktu (s)	
	Ponsel	Emulator
1	2.08	3.31
2	2.17	3.19
3	2.27	3.01
4	2.36	2.95
5	2.38	3.38
6	2.40	2.57

Percobaan ke -	Waktu (s)	
	Ponsel	Emulator
7	2.27	2.85
8	2.13	2.83
9	2.20	3.10
10	2.47	2.76



Gambar 4.5 Grafik Pengujian $\text{fps} = 1000/70$

Berdasarkan pada pengukuran dengan $\text{fps} = 1000/70$ didapat waktu saat karakter muncul samapi karakter menghilang, dapat disimpulkan bahwasanya rata – rata pemain dalam mengakses aplikasi game pada ponsel selama tersebut selama 2.273s. Sedangkan pada emulator hasil rata – ratanya 2.995s.

Dari keseluruhan pengujian diatas didapat bahwa semua pengujian pertama dengan menggunakan emulator rata – rata mengalami waktu yang lebih lama. Hal ini dikarenakan emulator Android yang memakan *memory* PC yang besar. Sedangkan pada ponsel terjadi kestabilan dalam memainkan aplikasi ini namun untuk memainkan waktu yang lama, dikhawatirkan terjadi *error* pada ponsel dikarenakan aplikasi akan terus *standby* sesuai dengan siklus Android pada gambar 2.11.

5. KESIMPULAN

Dari hasil pembuatan dan pengukuran sistem didapat beberapa kesimpulan bahwa.

- Aplikasi ini selalu standby jika ditutup, melihat siklus alur Android pada gambar 2.1.
- Menggunakan random pada *looping* karakter berefek pada ketidak pastian keluaran pada karakter.
- Akibatnya waktu saat karakter keluar dan musik mengeluarkan suara, tidak bersamaan.
- Bertumbuhkannya karakter tidak dapat dipungkiri lagi, namun dapat diminimalisir.
- Kelebihan format audio midi adalah ukuran file kecil dan ringan saat diakses oleh sistem operasi Android.

- Kekurangan format audio midi adalah volume yang didengar oleh pemain aplikasi, dirasa masih kecil.
- Semakin kecil fps maka waktu rata – rata yang dihasilkan akan semakin cepat.

6. DAFTAR PUSTAKA

- [1] Haseman, Chris (2008), *Android Essentials*. New York: Springer-Verlag
- [2] Zechner, Mario (2009). *Beginning Android Games*. New York: Springer-Verlag
- [3] Silva, Vladimir (2009). *Pro Android Games*. New York: Springer-Verlag
- [4] Murphy, Mark L. (2009). *Beginning Android*. New York: Springer-Verlag
- [5] Murphy, Mark L. (2010). *Beginning Android 2*. New York: Springer-Verlag
- [6] Android Developers. 2010. *Android SDK Docs*. <http://www.android.com/>. Diakses tanggal 28 Juni 2011